Software Population Pyramids: The Current and the Future of OSS Development Communities

Saya Onoue, Hideaki Hata, Kenichi Matsumoto
Graduate School of Information Science
Nara Institute of Science of Technology, Nara, Japan
{onoue.saya.og0, hata, matumoto}@is.naist.jp

ABSTRACT

Context: Since human power is an essential resource, the number of contributors in a software development community is one of the health indicators of an open source software (OSS) project. For maintaining and increasing the populations in software development communities, both attracting new contributors and retaining existing contributors are important. **Goal**: Our goal is understanding the current status of projects' population, especially the different experienced contributors' composition of the projects. **Method**: We propose software population pyramids, a graphical illustration of the distribution of various experience groups in a software development community. **Results**: From the study with OSS projects in GitHub, we found that the shapes of software population pyramids varies depending on the current status of OSS development communities. **Conclusions**: This paper present a software population pyramid of the distribution of various experience groups in a software community population. Our results can be considered as predictors of the near future of a project.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems—*human factors*; J.4 [**Social and Behavioral Sciences**]: Sociology

General Terms

Human Factors

Keywords

OSS, software development community, population pyramid

1. INTRODUCTION

The Heartbleed security vulnerability in OpenSSL clarified the challenges of open source software (OSS) projects, that is, the population of software development communities. In the OpenSSL project, there had been only a few full-time core developers and a number of part-time volunteer developers, and this shortage of human resources can be considered as one of the causes[1]. Therefore maintaining and increasing the populations in software development communities are one of the challenges of OSS projects.

Zhou and Mockus studied long term contributors (LTC), by addressing the research question "what impacts the chances that a new joiner to a software project will become an LTC[5]?" From the analysis of the behavior of individual participants in Gnome and Mozilla, they reported that future LTCs tend to be more active and show more community-oriented attitude than other joiners during their first month.

Yamashita et al. proposed a pair of population metrics, namely, magnetism and stickiness[4]. They defined magnet projects as those that attract a large proportion of new contributors, and sticky projects as those where a large proportion of the contributors will keep making contributions. With two values of magnetism and stickiness, OSS projects are classified into the following four categories. **Attractive** projects have high magnet and high sticky values. These projects are successful in both attracting new contributors and retaining existing ones. **Fluctuating** projects have high magnet but low sticky values. These projects are successful in attracting new contributors, but unsuccessful in retaining them. So the members of these OSS development communities fluctuate year by year. **Stagnant** projects have low magnet but high sticky values. These projects are contrary to the fluctuating projects, that is, they retain the existing contributors but cannot attract new ones. **Terminal** projects have low magnet and low sticky values. Based on this classification, they empirically studied OSS project histories, and found at-risk projects.

To see the population of OSS development communities in detail, we adopt population pyramids, a graphical illustration of the distribution of various age groups in a population. In general, population pyramids are used to show the current status of countries' population, and may provide insights about political and social stability, as well as economic development. In addition, they are known to be powerful predictors of the future[2]. Depending on the countries' situations, the shapes of population pyramids varies. In our

ESEM '14 September 18-19, 2014, Torino, Italy.

[1]Free Can Make You Bleed `http://www.ssh.com/blog/makesyoubleed`, Apr. 30, 2014.
[2]Population pyramids: Powerful predictors of the future - Kim Preshoff, TED-Ed, http://ed.ted.com/lessons/population-pyramids-powerful-predictors-of-the-future-kim-preshoff

software population pyramids, contributors are grouped by their experiences in the communities. With the same data of the previous study[4], we created software population pyramids for OSS project communities and analyzed them. The differences between our study and the previous study can be summarized as follows:

- Yamashita et al. considered a developer to be authors of code changes. So they only focused on the commit and pull request activities. However, we are also interested in other contributors who send issues and comments. So we analyze other activities as well as commit and pull request activities.
- Our software population pyramids are consisted of various experience groups in a software development communities. So, our method can see LTCs, though the previous study didn't distinguish the developers' experiences.

We found that (i) the shapes of software population pyramids varies depending on the current status of OSS project communities (Attractive, Fluctuating, Stagnant, and Terminal), and (ii) software population pyramids can be considered as predictors of the near future.

2. POPULATION PYRAMIDS FOR OSS

In a general population pyramid, the population is distributed along the horizontal axis, with males shown on the left and females on the right. The male and female populations are broken down into 5-year age groups represented as horizontal bars along the vertical axis, with the youngest age groups at the bottom and the oldest at the top. The shape of the population pyramid gradually evolves over time based on fertility, mortality, and international migration trends.

In this paper we propose software population pyramids, population pyramids of software development communities. Contributors are considered as the constituent member of the communities, and the contribution periods are regarded as existing periods or lifetimes. Although the previous study limited contributions to coding activities[4], we treat both coding activities and discussion activities as contributions to see the volumes of overall contributors' populations. There are some differences between our software population pyramids and the general population pyramids.

- A population pyramid consists of males' and females' bars. But a software population pyramid consists of coding contributors' and discussion contributors' bars.
- In a general population pyramid, people appear at birth and disappear when they die. But in a software population pyramid, contributors start their experiences when they join and end when they leave the development communities.
- The height of the population pyramids are similar each other because people do not live more than 200 years. But software population pyramids have different height since OSS projects have different existing periods and people can leave freely.
- Since the parent-child relationships exist in population pyramids, there are correlation between the volume of parent population and children population. However, software population pyramid do not have such relationships. This can cause the pyramid change dramatically.

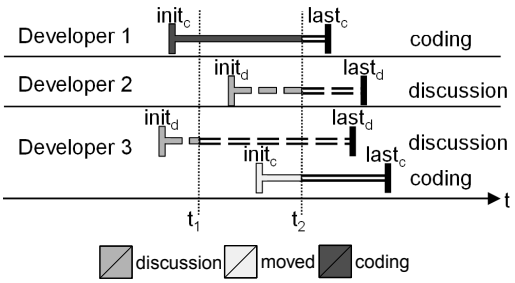


Figure 1: Contribution periods.

3. DATA COLLECTION

Similar to the previous study[4], we analyze the GitHub dataset provided by Gousios[3]. This dataset includes developers' activity histories of 90 OSS projects. We classified them into coding and discussion contributions. **Coding** contributions include commits and pull_requests. **Discussion** contributions include commit_comments, issue_comments, pull_request_comments, and issue_events.

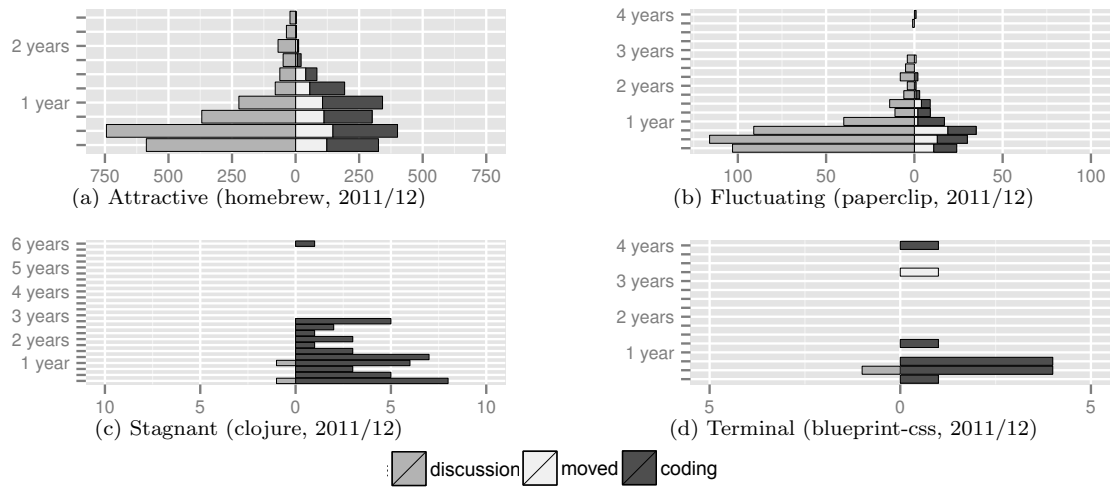
We obtained the dates of those events for each contributor, and identify the contribution period from the first event to the last event. Contribution periods are divided into **coding periods** and **discussion periods** based on the classification of the activity events. In Figure 1, the coding periods are represented as solid lines, and the discussion periods are represented as dotted lines. At the time t_1, there are two contributors. Developer 1 has the coding experience from $init_c$ to t_1, and developer 3 has the discussion experience from $init_d$ to t_1. In our software population pyramids, contributors with discussion experiences are shown on the left, and contributors with coding experiences are shown on the right.

Since coding is the essential contribution in OSS development, we distinguish coding contributions from other discussion contributions, and present separately in software population pyramids. Although some contributors only work on discussion, there are chances that contributors move from discussion contributions to coding contributions. At the time t_2, there are three contributors. Developer 1 has continued the coding contribution, and developer 2 starts discussion contributions. Developer 3, who have contributed on discussion, stared coding contributions before t_2. We consider him or her as a coding contributor with the experience from $init_c$ to t_2, and marked as **moved** contributors. Coding periods are represented as gray, discussion periods are represented as dark gray, and moved periods are represented as light gray.

4. ANALYSIS

We present the results of the analysis using software population pyramids with respect to two research questions.

RQ1: Are there typical shapes of software population pyramids depending on the current status of OSS development communities?



(a) Attractive (homebrew, 2011/12)
(b) Fluctuating (paperclip, 2011/12)
(c) Stagnant (clojure, 2011/12)
(d) Terminal (blueprint-css, 2011/12)

Figure 2: Software population pyramids and the current status.

In the previous study, the magnetism of a project is calculated as the proportion of contributors who made their first contribution in the time period, and the stickiness of a project is calculated as the proportion of contributors in the time period who have also made contributions in the following time period[4]. They empirically classified projects into four categories: attractive (high magnet and high sticky), fluctuating (high magnet and low sticky), stagnant (low magnet and high sticky), and terminal (low magnet and low sticky). Note that these classifications are based only on the number of coding contributors. The previous study did not take discussion contributions into account for the current status typology of OSS development communities. We present four software population pyramids belonging to such four categories in Figure 2.

(a) The project `homebrew` on December 2011 is *attractive*. Attractive projects have both new and experienced contributors. In the software population pyramid, we see many coding contributors on right. Although the homebrew project has the largest magnet value in the dataset, its sticky value is not so high[4]. This can be seen in the pyramid, that is, there is much volume near the bottom and less volume near the top. In addition, we found there are many discussion contributors on left comparable to the coding contributors. As a result, the pyramid forms balanced shape. Other attractive projects have similar balanced shape.

(b) The project `paperclip` on December 2011 is *fluctuating*, many contributors come and leave. First, we found there is much volume of coding contributors near the bottom and less volume near the top. Second, we found that there are many discussion contributors compared to the coding contributors. This project seems to be successful in attracting new contributors including both coding and discussion contributors, but unsuccessful in retaining them, especially retaining coding contributors. The fluctuating projects found to have left-sided pyramids.

(c) The project `clojure` on December 2011 is *stagnant*. The majority of the community members are coding contributors. Although there are many valuable coding contributors, there are little discussion contributors. These projects are contrary to the fluctuating projects, that is, the stagnant projects have right-sided pyramids.

(d) The project `blueprint-css` on December 2011 is *terminal*. There are less coding contributors and discussion contributors. The shapes of the terminal projects' software population pyramids are collapsed.

RQ2: How do software population pyramids change over time?

We investigated the transition of software population pyramids over time. Figure 3 shows the transition of the software population pyramids of three OSS projects. Software population pyramids are created at the four snapshot on June 2010, June 2011, June 2012, and June 2013.

The `homebrew` project has been *attractive* since June 2011. On June 2010, there are many discussion contributors, but not many coding contributors. Until June 2011, it attract many coding contributors as well as discussion contributors. From then, the homebrew development community grows. It keeps attracting new contributors, and retain existing contributors. As a result, the height of the pyramid increased, and the shape becomes like pyramid.

Next, we examine the changes of the `blueprint-css`'s software population pyramids, which was a *terminal* project in 2011. The shape of this software population pyramid is unbound and unstable. This project had little discussion contributors from 2010 to 2013, and do not have them in 2013. Long term developers who have stayed until 2011 disappeared in 2012. Current contributors of this project are different from the previous contributors.

Finally, we examine the changes of the `jekyll`'s software population pyramids, which was also a *terminal* project in 2011. This software population pyramid has number of discussion contributors more than number of coding contributors. After that, it has a little moved developers, and number of both contributors increased in 2011. Number of discussion and coding contributors continued to increase, and

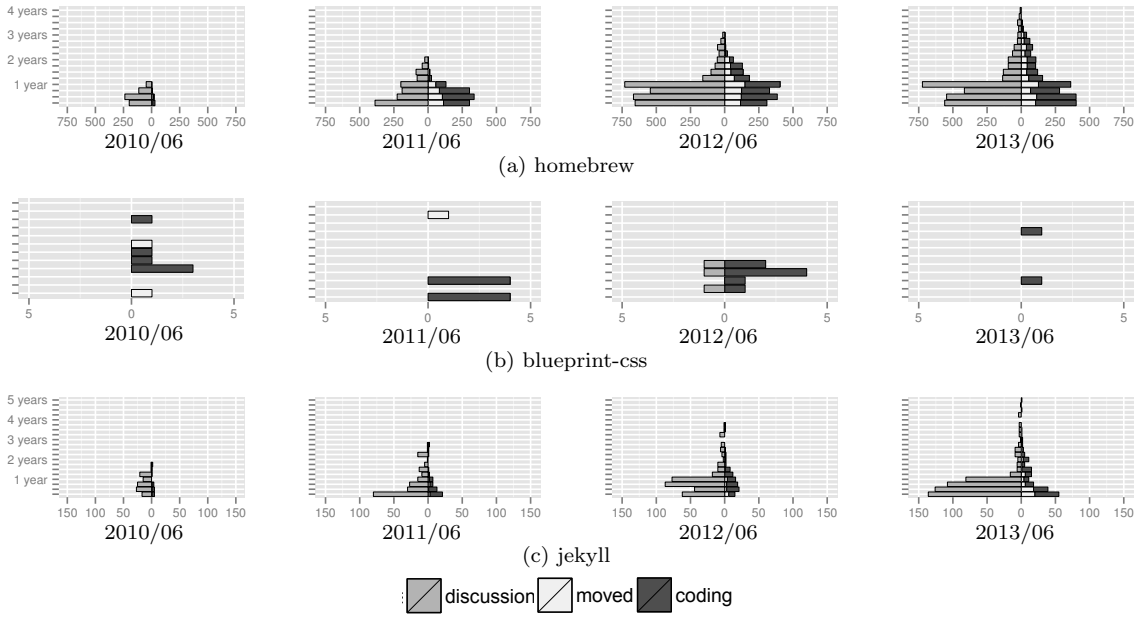


Figure 3: The transitions of software population pyramids.

the population pyramid becomes balanced shape in 2013. This project classified as terminal project in 2011. However, there are many discussion contributors, which is different from the case of the blueprint-css. Therefore, we think this project had a possibility to become attractive or fluctuating project in near future.

In summary, we see that the population of the current status can be predictors of near future. Especially, discussion contributors are important to attract new coding contributors, and they may be potential coding contributors.

5. RELATED WORK

Human factor is a related field of study to this research. For example, Zhou and Mockus et al. Reported that a new contributors to become a Long Term Contributor (LTC) tend to show more community-oriented attitude than other contributors[5]. Our study presented entry and exit of contributors of OSS projects. These results indicated that participation and retirement of contributors are different to every project.

Social Network Analysis is a related field of study to this research. For example, Bird et al. Reported that developers play a significant social role in email lists [1]. Similarly, Bird et al analyzed email addresses in open source software projects to examine the community structure among developers [2]. Although our study collected data from different kinds of development archives, specifically the developers' activity events in GitHub, these results also indicated contributors have many different roles.

6. CONCLUSIONS

This paper presented a graphical illustration of the distribution of various experience groups in a software community population, called a software population pyramid. Software population pyramids show the volumes of contributors in the software development communities. We distinguish coders from not coding contributors, and designed the software population pyramids, to see the transition from not coding contributions to coding contributions. From an empirical study with OSS projects in GitHub, we found that the shapes of software population pyramids vary depending on the current status of development communities. We think that lively discussions may attract new contributors, and lively developments may agitate existing contributors for staying. Clarifying such effects is one of our future work, and we plan to continue studying developers' migration over OSS projects.

7. ACKNOWLEDGMENTS

This study has been supported by JSPS KAKENHI Grant Number 26540029, and has been conducted as a part of "Research Initiative on Advanced Software Engineering in 2013" supported by Software Reliability Enhancement Center (SEC), Information Technology Promotion Agency Japan (IPA).

8. REFERENCES

[1] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan. Mining email social networks. In *Proc. of* MSR '06, pp. 137–143, 2006.

[2] C. Bird, D. Pattison, R. D'Souza, V. Filkov, and P. Devanbu. Latent social structure in open source projects. In *Proc. of* SIGSOFT '08/FSE-16, pp. 24–35, 2008.

[3] G. Gousios. The ghtorent dataset and tool suite. In *Proc. of* MSR '13, pp. 233–236, 2013.

[4] K. Yamashita, S. McIntosh, Y. Kamei, and N. Ubayashi. Magnet or sticky? an oss project-by-project typology. In *Proc. of* MSR '14, pp. 344–347, 2014.

[5] M. Zhou and A. Mockus. What make long term contributors: Willingness and opportunity in oss community. In *Proc. of* ICSE '12, pp. 518–528, 2012.